

Problem C. Even String

Time limit 1000 ms

Mem limit 262144 kB

A string $a = a_1a_2 \dots a_n$ is called *even* if it consists of a concatenation (joining) of strings of length 2 consisting of the same characters. In other words, a string a is even if two conditions are satisfied **at the same time**:

- its length n is even;
- for all odd i ($1 \leq i \leq n - 1$), $a_i = a_{i+1}$ is satisfied.

For example, the following strings are even: "" (empty string), "tt", "aabb", "oooo", and "ttrrrroouuuuuuuukk". The following strings are not even: "aaa", "abab" and "abba".

Given a string s consisting of lowercase Latin letters. Find the minimum number of characters to remove from the string s to make it even. The deleted characters do not have to be consecutive.

Input

The first line of input data contains an integer t ($1 \leq t \leq 10^4$) — the number of test cases in the test.

The descriptions of the test cases follow.

Each test case consists of one string s ($1 \leq |s| \leq 2 \cdot 10^5$), where $|s|$ — the length of the string s . The string consists of lowercase Latin letters.

It is guaranteed that the sum of $|s|$ on all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, print a single number — the minimum number of characters that must be removed to make s even.

Examples

Input	Output
6	3
aabbdabdbcc	3
zyx	2
aaababbb	0
aabbcc	2
oaoaaaoo	7
bme fbmuyw	

Note

In the first test case you can remove the characters with indices 6, 7, and 9 to get an even string "aabbddcc".

In the second test case, each character occurs exactly once, so in order to get an even string, you must remove all characters from the string.

In the third test case, you can get an even string "aaaabb" by removing, for example, 4-th and 6-th characters, or a string "aabbabb" by removing the 5-th character and any of the first three.